

Software Group

Compilation Technology

IBM System p Compiler Roadmap

Roch Archambault
IBM Toronto Laboratory
archie@ca.ibm.com

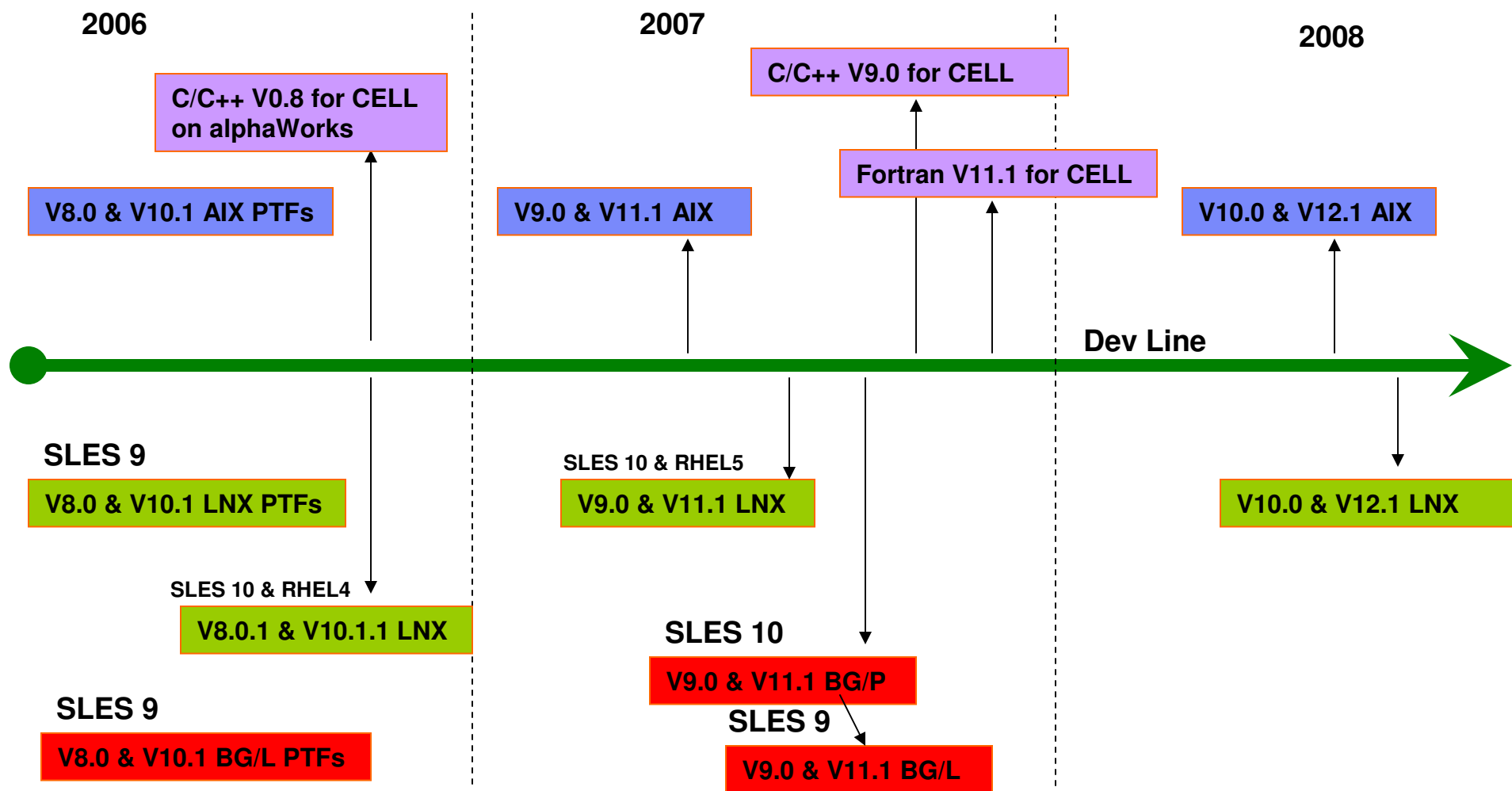


Agenda

- Overall Roadmap
- The System p Compiler Products
- Detailed Roadmaps
 - Common Features & Compiler Architecture
 - XL Fortran
 - XL C/C++
 - XL Compilers for Blue Gene
 - XL Compilers for Cell
 - XL UPC Compiler
- Customer Requirements
- Online documentation
- Performance Comparison
- Q&A



Roadmap of XL Compiler Releases



All information subject to change without notice



The System p Compiler Products: Latest Versions

- **All POWER4, POWER5, POWER5+ and PPC970 enabled**

- XL C/C++ Enterprise Edition V8.0 for AIX

- XL Fortran Enterprise Edition V10.1 for AIX

- XL C/C++ Advanced Edition V8.0 for Linux (SLES 9 & RHEL4)

- XL Fortran Advanced Edition V10.1 for Linux (SLES 9 & RHEL4)

- XL C/C++ Advanced Edition V8.0.1 for Linux (SLES 10 & RHEL4)

- XL Fortran Advanced Edition V10.1.1 for Linux (SLES 10 & RHEL4)

- XL C/C++ Enterprise Edition for AIX, V9.0 (**POWER6 enabled**)

- XL Fortran Enterprise Edition for AIX, V11.1 (**POWER6 enabled**)

- **Blue Gene/L (BG/L) enabled**

- XL C/C++ Advanced Edition V8.0 for BG/L (PRPQ)

- XL Fortran Advanced Edition V10.1 for BG/L (PRPQ)



The System p Compiler Products: Latest Versions

- **Technology Preview currently available from alphaWorks**

IBM XL C/C++ Alpha Edition for Cell Broadband Engine Processor on Linux,
V0.8.1 (support SDK 2.0 on FC5 x86 and PPC)

IBM XL C/C++ Alpha Edition for Cell Broadband Engine Processor on Linux,
V0.8.2 (support SDK 2.1 on FC6 x86 and PPC)

Download: <http://www.alphaworks.ibm.com/tech/cellcompiler>

IBM XL Fortran Alpha Edition for Cell Broadband Engine Processor on Linux,
V0.11

Download: <http://www.alphaworks.ibm.com/tech/cellfortran>

XL UPC language support on AIX and Linux

Download: <http://www.alphaworks.ibm.com/tech/upccompiler>



The System p Compiler Products: End Of Service

- The following compilers went out of service in April 2007:
 - VisualAge C++ Version 6.0 for AIX
 - VisualAge C++ Version 6.0 for Linux
 - C for AIX V 6.0
 - XL Fortran Version 8.1 for AIX
 - XL Fortran Version 8.1 for Linux



The System p Compiler Products: Future Versions

- POWER6 enabled Linux releases
 - XL C/C++ Advanced Edition for Linux, V9.0
 - XL Fortran Advanced Edition for Linux, V11.1
 - Blue Gene/P (BG/P) enabled
 - XL C/C++ Advanced Edition for BG/P, V9.0
 - XL Fortran Advanced Edition for BG/P, V11.1
- Blue Gene/L (BG/L) enabled
 - XL C/C++ Advanced Edition for BG/L, V9.0
 - XL Fortran Advanced Edition for BG/L, V11.1

All information subject to change without notice



The System p Compiler Products: Future Versions

- Cell/B.E. cross compiler products:
 - XL C/C++ for Cell SDK 3.0 Linux, V9.0 (x86 and PPC)
 - XL Fortran for Cell SDK 3.0 Linux, V11.1 (PPC only)
- Cell/B.E. cross compilers from alphaWorks:
 - XL C/C++ for Cell SDK 3.0 Linux, V0.9.1 (x86 and PPC)
 - (Tech preview for single source compiler)
- Extra POWER6 performance
 - XL C/C++ Enterprise Edition for AIX, V10.0
 - XL Fortran Enterprise Edition for AIX, V12.1
 - XL C/C++ Advanced Edition for Linux, V10.0
 - XL Fortran Advanced Edition for Linux, V12.1

All information subject to change without notice



Common Fortran, C and C++ Features

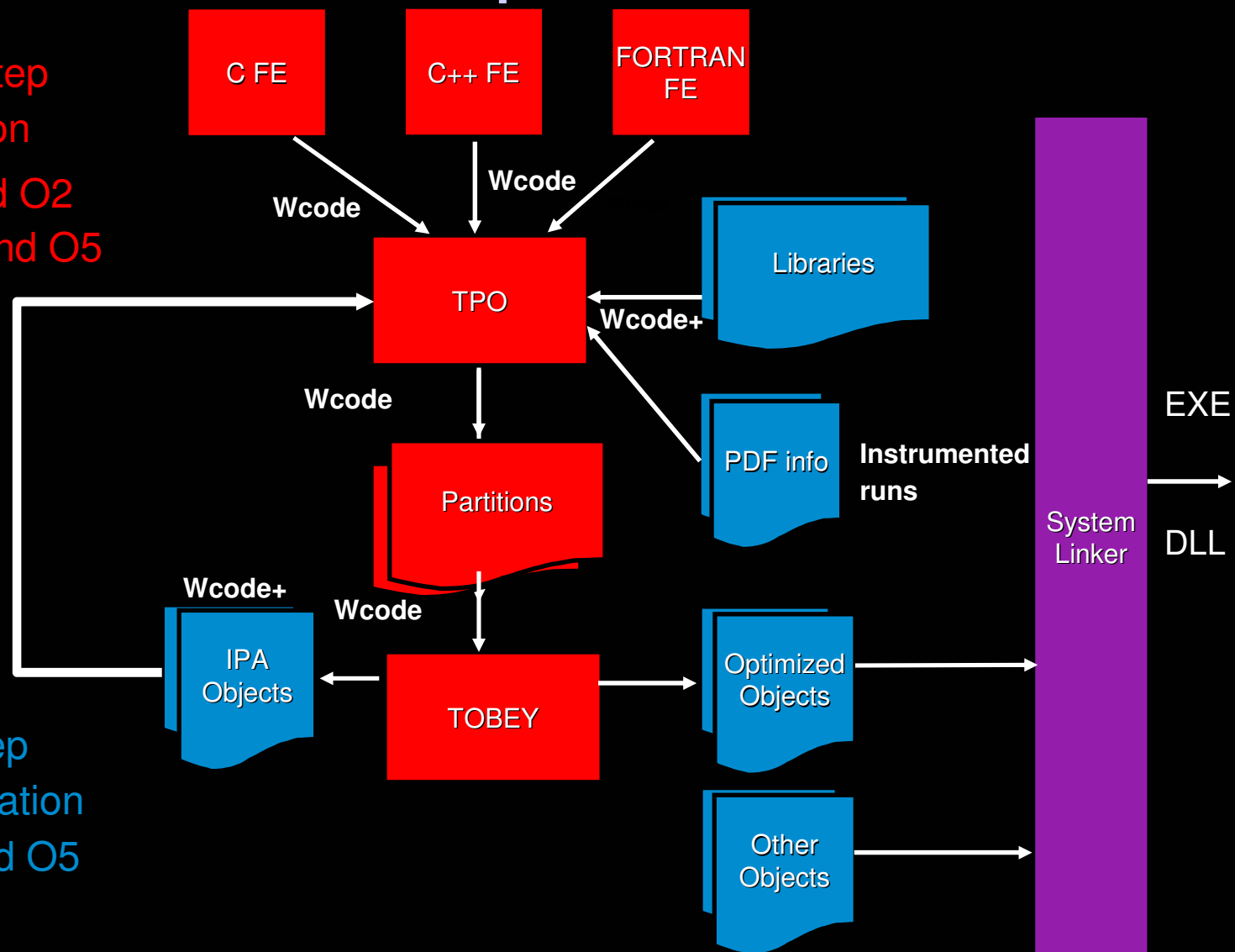
- Linux (SLES and RHEL) and AIX, 32 and 64 bit
- Debug support
 - Debuggers on AIX:
 - Total View (TotalView Technologies), DDT (Allinea), IBM Debugger and DBX
 - Debuggers on Linux:
 - TotalView, DDT and GDB
- Full support for debugging of OpenMP programs (TotalView)
- Snapshot directive for debugging optimized code
- Portfolio of optimizing transformations
 - Instruction path length reduction
 - Whole program analysis
 - Loop optimization for parallelism, locality and instruction scheduling
 - Use profile directed feedback (PDF) in most optimizations
- Tuned performance on POWER3, POWER4, POWER5, PPC970, PPC440, POWER6 and CELL systems
- Optimized OpenMP



IBM XL Compiler Architecture

Compile Step
Optimization

noopt and O2
O3, O4 and O5





XL Fortran Roadmap: Strategic Priorities

- **Superior Customer Service**

Continue to work closely with key ISVs and customers in scientific and technical computing industries

- **Compliance to Language Standards and Industry Specifications**

OpenMP API V2.5

Fortran 77, 90 and 95 standards

Fortran 2003 Standard

- **Exploitation of Hardware**

Committed to maximum performance on POWER4, PPC970, POWER5, POWER6, PPC440 and successors

Continue to work very closely with processor design teams



XL Fortran Version 11.1 for AIX/Linux – Spring/Summer 2007

- AIX Announcement Letter:
<http://www.ibm.com/common/ssi/cgi-bin/ssialias?infotype=an&subtype=ca&appname=Demonstration&htmlfid=897/ENUS207-125>
- Continued rollout of Fortran 2003
- Compliant to OpenMP V2.5
- Perform subset of loop transformations at –O3 optimization level
- Tuned BLAS routines (DGEMM and DGEMV) are included in compiler runtime (libxlopt)
- Recognize matrix multiply and replace with call to DGEMM
- Runtime check for availability of ESSL
- Support for auto-simdization and VMX intrinsics (and data types) on AIX
- Inline MASS library functions (math functions)



XLF 11.1 delivers most of the remaining F2003 standard

- Full support of procedure pointers and allocatable object semantics
- Object-oriented Fortran programming with constructs similar to C++ classes, methods, and destructors
- User-defined derived type I/O
- Derived Type Parameters (similar to C++ templates) will be the only major F2003 feature not available in 11.1



XL C/C++ Roadmap: Strategic Priorities

- **Superior Customer Service**
- **Compliance to Language Standards and Industry Specifications**
 - ANSI / ISO C and C++ Standards
 - OpenMP API V2.5
- **Exploitation of Hardware**
 - Committed to maximum performance on POWER4, PPC970, POWER5, PPC440, POWER6 and successors
 - Continue to work very closely with processor design teams
- **Exploitation of OS and Middleware**
 - Synergies with operating system and middleware ISVs (performance, specialized function)
 - Committed to AIX Linux affinity strategy and to Linux on pSeries
- **Reduced Emphasis on Proprietary Tooling**
 - Affinity with GNU toolchain



XL C/C++ Version 9.0 for AIX/Linux – Spring/Summer 2007

- AIX Announcement Letter:

[http://www.ibm.com/common/ssi/fcgi-](http://www.ibm.com/common/ssi/fcgi-bin/ssialias?infotype=an&subtype=ca&appname=Demonstration&htmlfid=897/ENUS207-124)

[bin/ssialias?infotype=an&subtype=ca&appname=Demonstration&htmlfid=897/ENUS207-124](http://www.ibm.com/common/ssi/fcgi-bin/ssialias?infotype=an&subtype=ca&appname=Demonstration&htmlfid=897/ENUS207-124)

- Compliant to OpenMP V2.5
- Perform subset of loop transformations at –O3 optimization level
- Tuned BLAS routines (DGEMM and DGEMV) are included in compiler runtime (libxlopt)
- Recognize matrix multiply and replace with call to DGEMM
- Runtime check for availability of ESSL
- Support for auto-simdization and VMX intrinsics on AIX
- Inline MASS library functions (math functions)
- Exploit “restrict” keyword in C 1999
- Partial compliance to C++ TR1 libraries and Boost 1.34.0
- Support for -qtemplatedepth which allows the user to control number of recursive template instantiations allowed by the compiler.
- Exploit DFP and VMX on Power6.
- Improved inline assembler support



Blue Gene Compilers

XL C/C++ Advanced Edition V8.0 for BG/L and XL Fortran Advanced Edition V10.1 for BG/L

- Performance tuning of SPEC2000FP, DDCMD Kernels, NAS 3.2 Serial and sPPM.
- Performance tuning of MASS library
- Exploit 440D instructions for complex arithmetic
- BG/L compiler white paper (Exploiting the Dual FPU in BG/L):
<http://www.ibm.com/support/docview.wss?uid=swg27007511>
- June 2006 PTF (compiler refresh):
 - Support Blue Gene software release 3
 - Overall SPEC2000FP faster for 440D than 440
 - Updated white paper to reflect June 2006 PTF performance improvements
- December 2006 PTF (compiler refresh)
 - Continue to improve 440D performance of benchmarks listed above
 - Updated white paper to reflect December 2006 PTF performance improvements



Blue Gene Compilers

XL C/C++ Advanced Edition for BG/P, V9.0 and XL Fortran Advanced Edition for BG/P, V11.1

- Support for OpenMP, automatic parallelization and dynamic linking
- Performance improvements: SIMD and other general optimizations
- MASS/MASSV performance improvements
- FEN (Front End Node) is SLES10

XL C/C++ Advanced Edition for BG/L, V9.0 and XL Fortran Advanced Edition for BG/L, V11.1

- Same code base as BG/P release except FEN is SLES 9
- GA is one month after BG/P GA

All information subject to change without notice



Cell/B.E. Compilers

- **Currently available on alphaWorks**
- **IBM XL C/C++ Alpha Edition for Cell Broadband Engine Processor on Linux, V0.8.1**
 - Hosted on Linux x86 and Linux PPC (FC5)
 - Support SDK 2.0 interfaces
 - Targets QS20 Blade (Cell Blade 1 hardware)
- **IBM XL C/C++ Alpha Edition for Cell Broadband Engine Processor on Linux, V0.8.2**
- **IBM XL Fortran Alpha Edition for Cell Broadband Engine Processor on Linux, V0.11**
 - Hosted on Linux x86 (C/C++ only) and Linux PPC (FC6)
 - Support SDK 2.1 interfaces
 - Targets QS20 Blade
 - Support QS22 (Cell Blade 2 hardware) via simulator



Cell/B.E. Compilers

- **Future cross compilers products:**

- Hosted on RHEL5U1 (Red Hat) and F7 (Fedora)

- Hosted on x86 and PPC (separate products)

- Support SDK 3.0 interfaces

- Targets QS20 and QS21 Blades

- Targets QS22 Blade (Soma Hardware or Cell Blade 2)

- IBM XL C/C++ Advanced Edition for Multicore Acceleration for Linux, V9.0

- IBM XL Fortran Advanced Edition for Multicore Acceleration for Linux, V11.1

- **Future tech previews on alphaWorks:**

- Hosted on RHEL5U1 and F7

- Hosted on x86 and PPC

- Support SDK 3.0 interfaces

- User directed single source compiler (using OpenMP)

- IBM XL C/C++ alphaWorks Edition for Multicore Acceleration for Linux, V0.9.2

All information subject to change without notice



XL UPC Compiler

- **Future tech previews on alphaWorks**

- Based on XL C V9.0 compiler

- Compiler generated interface to the runtime system is identical for shared and distributed memory implementations

- Optimizations take advantage of system architecture knowledge

- **On AIX**

- Shared Memory (pthreads)

- Distributed (LAPI)

- **On Linux**

- Shared Memory (pthreads)

- Distributed (LAPI)

- **On BG/L**

- BG Message Layer

- **Using approximately 1000 test scenarios:**

- GWU UPC test suite

- UPC version of NAS benchmarks

- Berkeley UPC test suite

- MTU UPC test suite

- HPC Challenge suite

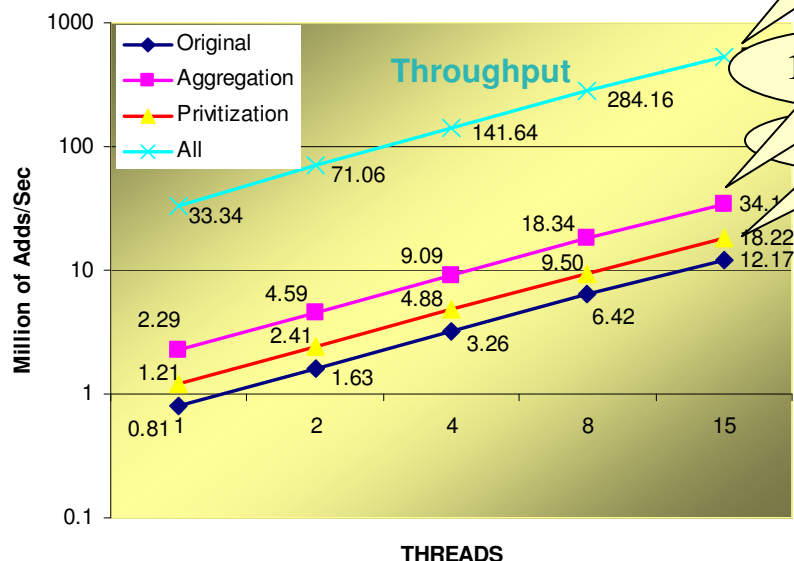
All information subject to change without notice



XL UPC Compiler: Optimization Improvements

```
shared [BF] int RES[N], V1[N], V2[N];
upc_forall (i=0; i<N; i++; &RES[i]) {
    RES[i] = V1[i+BF] + V2[i+BF];
}
```

```
for (i=BF*MYTHREAD, blk=0; i<N; i+=BF*THREADS, blk++)
{
    if (i%BF==0) { // aggregate V1 and V2
        upc_memget(&lv1[0], &V1[i+BF], BF*sizeof(int));
        upc_memget(&lv2[0], &V2[i+BF], BF*sizeof(int));
    }
    for (j=0; j<BF; j++) // privatize RES
        RES[(blk*BF)+j] = lv1[j] + lv2[j];
}
```



Working on the following optimizations :
(some will be available in tech preview)

- Parallel loop reshaping
- Reduce the overhead of local-shared accesses
- Aggregate fine grained communication in bulk transfers
- Schedule remote shared accesses to hide latency

All information subject to change without notice



Customer Requirements Shipped In 2006

- **Preserve User's CFG file (SPXXL 3.10) :**

Compiler reads a user config file specified using an environment variable (XLC_USR_CONFIG)

```
# /etc/vac.cfg
```

```
xlc:          use          = DEFLT
```

```
              options      = A B
```

```
DEFLT:        options      = C
```

```
# ./my.cfg
```

```
xlc:          use xlc
```

```
              options      = D
```

Compiler would use "D A B C" as options for the xlc stanza.

- **Separate environment variables for C/C++ (XLC_USR_CONFIG) and Fortran (XLF_USR_CONFIG)**
- **Shipped in August PTF for XL C/C++ V8.0 and XL Fortran V10.1**



Customer Requirements Shipped In 2007

■ Extended Support for `-qversion` option (SPXXL 3.9) :

```

+ --- noversion -----+
|                         |
>>-- -q version -+-----+><
|                         |
+--- = verbose -----+

```

- The default is `-qnoversion`. When `-qversion` is specified, the default is no suboption. If the option is specified more than once on the command line, the setting of the last one is taken.
- The verbose suboption asks the compiler to display the build level of each compiler phases. The information may look like the following:

```

XL Fortran Enterprise Edition for AIX, V11.1
Version: 11.01.0000.0001
Driver Version: 11.01 (Fortran) Level: 060414
Fortran Transformer Version: 11.01(Fortran) Level: 060419
Fortran Front End Version : 11.01(Fortran) Level: 060420
High Level Optimizer Version: 09.00(C/C++) and 11.01(Fortran) Level: 060411
Low Level Optimizer Version: 09.00(C/C++) and 11.01(Fortran) Level: 060418

```

- The `-qsaveopt` option will be extended to save the above information to the `.o` file.
- Shipped in XL C/C++ V9.0 and XL Fortran V11.1



Customer Requirements Shipped In 2007

- The following features are shipped in XL C/C++ V9.0 and XL Fortran V11.1 :
 - ✓ Provide Filename and Line Number in ALLOC/DEALLOC Failure (Fortran)
 - ✓ Provide Filename and Line Number in NAMELIST Failure (Fortran)
 - ✓ Little-Endian Data I/O Support (Fortran)
 - ✓ Thread Number in Standard Error output (Fortran)
 - ✓ Detect a thread's stack going beyond its limit (Fortran and C/C++). Implemented with –qsmp=stackcheck.
 - ✓ Initialize Allocatable Arrays with NaNs (Fortran). This will be done in AIX 6.1 via extension to malloc.
 - ✓ Improve performance of critical codes on BG/L
 - ✓ Support for –qtune=balanced for good performance on POWER6 without causing major degradation on POWER5
 - ✓ Compile time improvements for WRF (OpenMP with array section parms)
 - ✓ Improve reporting of automatic SIMDization (-qreport=hotlist)



New Compiler Options and Directives Shipped In 2007

- New suboptions to `–qfloat`:
 - `-qfloat=fenv` asserts that FPSCR may be accessed (default is `nofenv`)
 - `-qfloat=hscmplx` better performance for complex divide/abs (default is `nohscmplx`)
 - `-qfloat=nosingle` does not generate single precision float operations (default is `single`)
 - `-qfloat=norngchk` does not generate range check for software divide and (default is `rngchk`)
- `-qoptdebug` for debugging optimized code
- `-qxlf90=nosignedzero` now the default when `–qnostrict` (improves max/min perf)
- Builtin functions for new Power6 instructions:
 - `dcbfl` (*local flush*)
 - new dcbt variant (prefetch depth)*
 - `dcbst` (*store stream*)
- Expected value directive for function arguments.
- All the above has been shipped in XL C/C++ V9.0 and XL Fortran V11.1



Feature Request

- Request for a feature to be supported by our compilers
- C/C++ feature request page:
<http://www.ibm.com/support/docview.wss?uid=swg27005811>
- Fortran feature request page:
<http://www.ibm.com/support/docview.wss?uid=swg27005812>
- Or send e-mail to xl_feature@ca.ibm.com

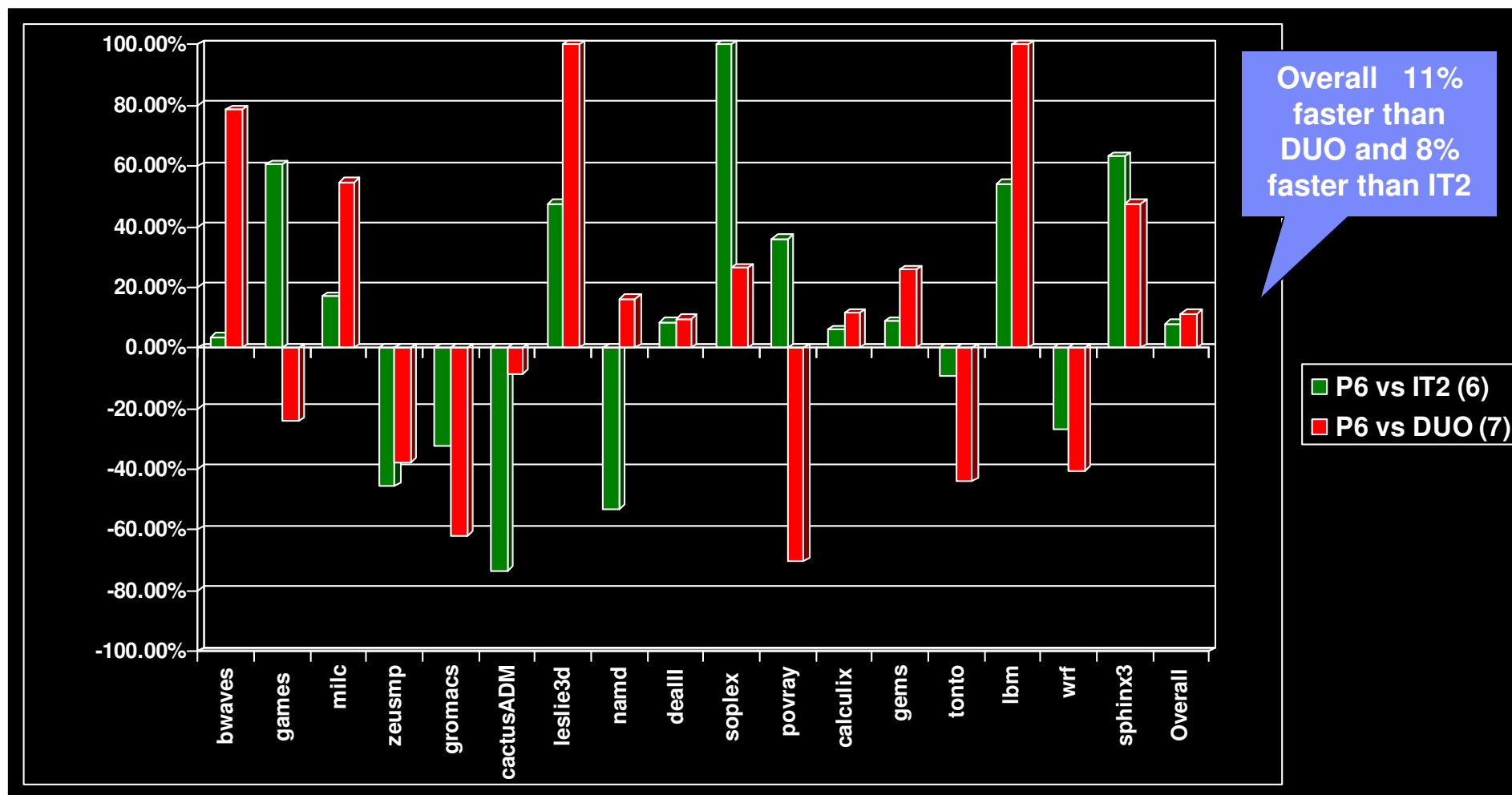


Documentation

- An information center containing the documentation for the **XL Fortran V10.1** and **XL C/C++ V8.0** versions of the AIX compilers is available at:
<http://publib.boulder.ibm.com/infocenter/comphelp/v8v101/index.jsp>
- An information center containing the documentation for the **XL Fortran V11.1** and **XL C/C++ V9.0** versions of the AIX compilers is available at:
<http://publib.boulder.ibm.com/infocenter/comphelp/v9v111/index.jsp>
- **Optimization and Tuning Guide** for XLF V10.1 and XLF V11.1 is now available online at: <http://publib.boulder.ibm.com/infocenter/comphelp/v9v111/index.jsp>
- New whitepaper “Overview of the IBM XL C/C++ and XL Fortran Compiler Family” available at: <http://www.ibm.com/support/docview.wss?uid=swg27005175>
- This information center contains all the html documentation shipped with the compilers. It is completely searchable.
- Please send any comments or suggestions on this information center or about the existing C, C++ or Fortran documentation shipped with the products to compinfo@ca.ibm.com.



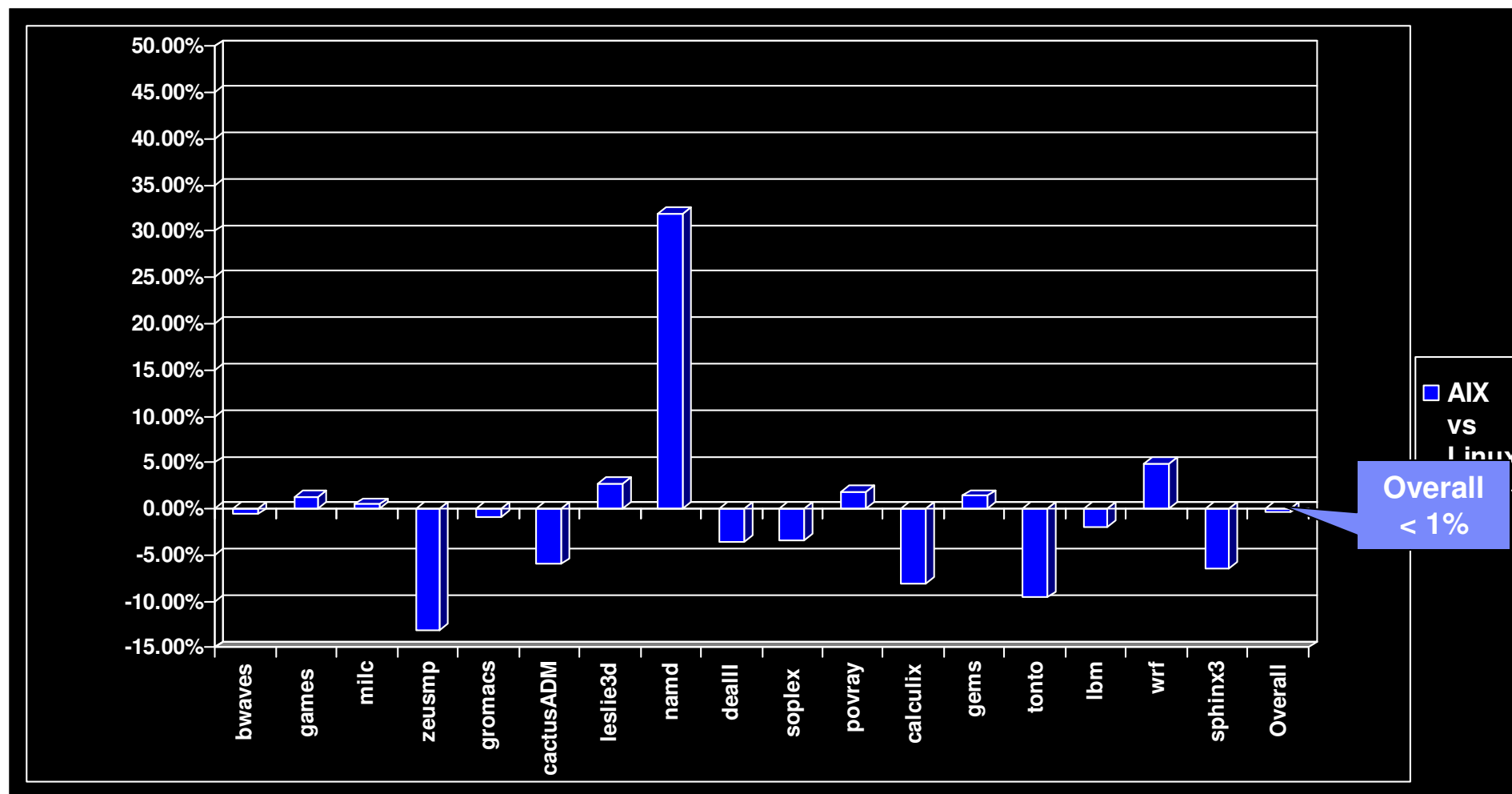
SPEC2006 FP Comparison Between Power6, Itanium-2 And Core Duo



Using base options from spec.org



SPEC2006 FP Comparison Between AIX and Linux on Power6



Using peak options from spec.org



BACKUP SLIDES



History Of Compiler Improvement On Power4

Compilers	2001 V5/V7.1.1	2002 V6/V8.1	2003 V6/V8.1.1	2004 V7/V9.1	2005 V8/V10.1	Compound Over 4 Years	CAGR Rate
SpecINT	baseline	21%	0%	3%	7%	34%	7.6%
SpecFLOAT	baseline	12%	5%	18%	5%	46%	9.9%

Note: SPEC2000 base options improvements from www.spec.org



Installation of Multiple Compiler Versions

- Installation of multiple compiler versions is supported
- The vacppndi and xlfndi scripts shipped with VisualAge C++ 6.0 and XL Fortran 8.1 and all subsequent releases allow the installation of a given compiler release or update into a non-default directory
- The configuration file can be used to direct compilation to a specific version of the compiler

Example: `xlf_v8r1 -c foo.f`

May direct compilation to use components in a non-default directory

- Care must be taken when multiple runtimes are installed on the same machine (details on next slide)



Coexistence of Multiple Compiler Runtimes

- **Backward compatibility**

C, C++ and Fortran runtimes support backward compatibility.

Executables generated by an earlier release of a compiler will work with a later version of the run-time environment.

- **Concurrent installation**

Multiple versions of a compiler and runtime environment can be installed on the same machine

Full support in xlfndi and vacppndi scripts is now available

- **Limited support for coexistence**

LIBPATH must be used to ensure that a compatible runtime version is used with a given executable

Only one runtime version can be used in a given process.

Renaming a compiler library is not allowed.

Take care in statically linking compiler libraries or in the use of *dlopen* or *load* .

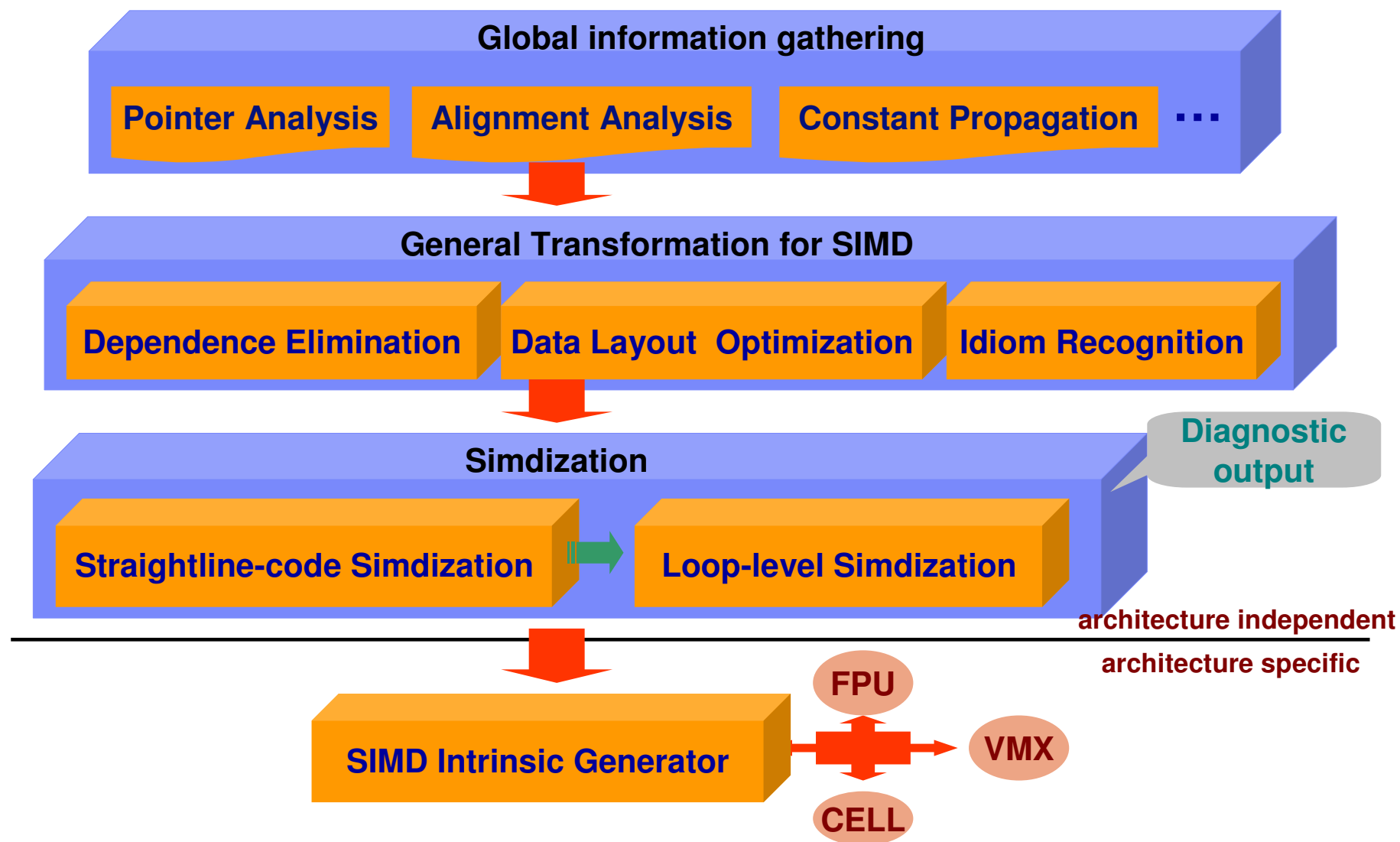
Details in the compiler FAQ

<http://www.ibm.com/software/awdtools/fortran/xlfortran/support/>

<http://www.ibm.com/software/awdtools/xlcpp/support/>



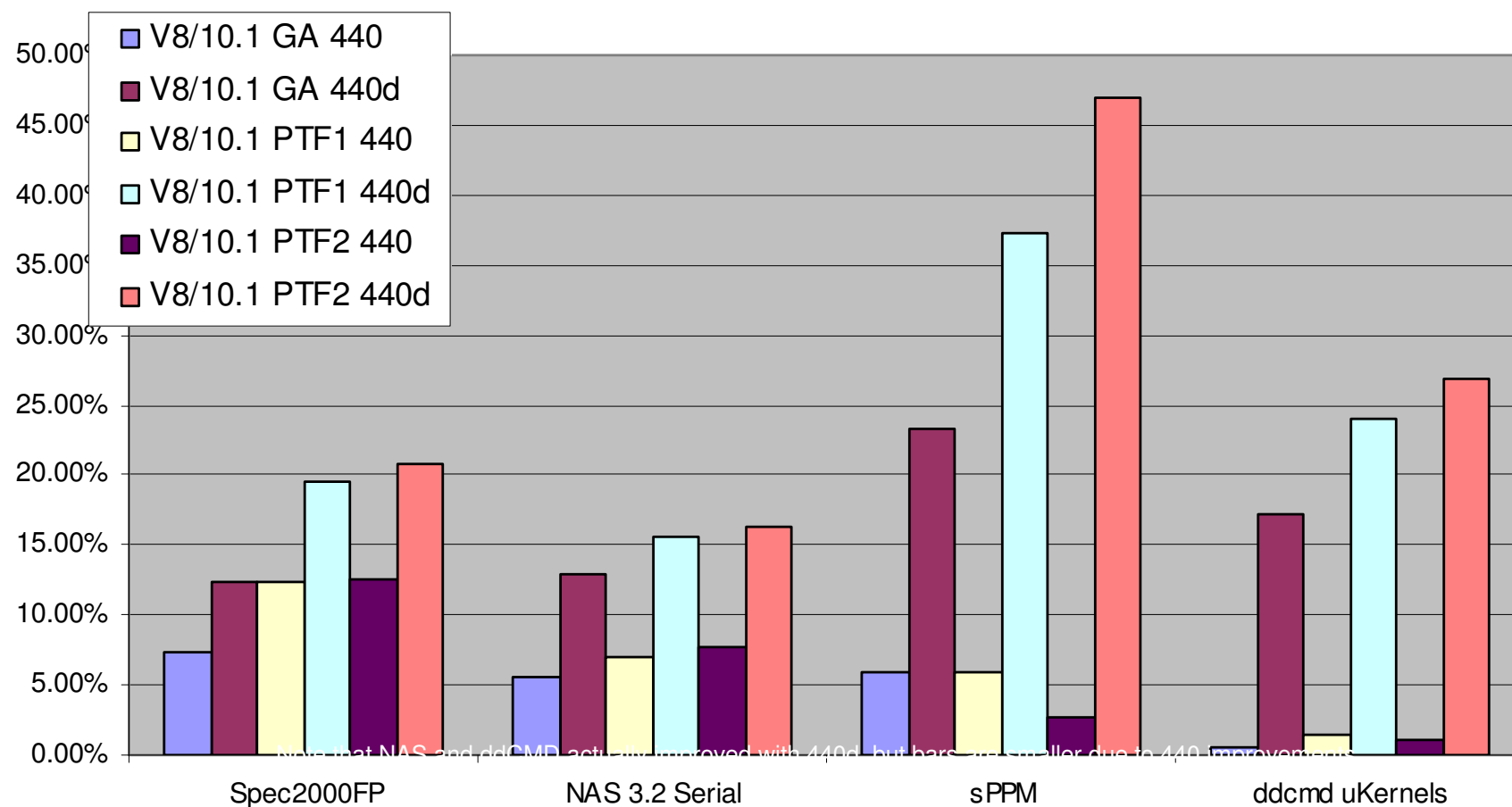
A Unified Simdization Framework





Blue Gene Compilers: Performance Results

Overall Improvement with -O5: V8/10.1 GA, PTF1, PTF2 vs. V7/9.1 Compilers





Blue Gene Compilers: Performance Results

**Overall Improvement with -O5:
-qarch=440d vs. -qarch=440**

